Amendment to the Claims

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Please amend the claims as follows:

1.   (Original) A method for tracing program flow within an application comprising:

providing options for modifying bytecode of the application at a plurality of levels of precision, each of the levels of precision specifying a particular set of methods of the application to be traced;

receiving a request to modify the bytecode at one of the of levels of precision;

responsively modifying the bytecode at the requested level of precision; executing the application; and

registering method invocations associated with the particular set of methods specified by the level of prevision.

2.   (Original) The method as in claim 1 wherein one of the levels of precision comprises all methods within the application.

3.   (Original) The method as in claim 1 wherein one of the levels of precision comprises all methods within a package of the application.

4.   (Currently amended) The method as in claim 3 wherein the application is a Java an object-oriented application and the package is a Java an object-oriented package.

5.   (Original) The method as in claim I wherein one of the levels of prevision comprises all methods within a particular class file of the application.

6.   (Original) The method as in claim 1 wherein one of the levels of prevision comprises individually identified methods of the application.

7.   (Original) The method as in claim 1 wherein modifying the bytecode comprises:
inserting a start method invocation prior to each method of the set of methods and inserting an end method invocation following each method of the set of methods.

8.   (Original) The method as in claim 7 further comprising:
storing method-related information associated with the method invocations of each of the particular set of methods.

9.   (Original) The method as in claim 7 wherein the method-related information comprises an amount of time it takes for each method within the set of methods to complete.

10. (Original) The method as in claim 7 wherein the method-related information comprises a number times that each method of the set of methods is executed.

11. (Original) The method as in claim 7 wherein the method-related information comprises input and/or output parameters associated with each method of the set of methods.

12. (Original) The method as in claim 9 further comprising:
constructing a hierarchical representation of the particular set of methods, the hierarchical representation including an indication of an amount of time it takes for each of the particular set of methods to complete.

13. (Original) The method as in claim 12 further comprising:
generating a graphical tree reflecting the hierarchical representation within a graphical user interface, wherein the graphical tree includes a separate entry for each method within the set of methods.

14. (Currently amended) A <u>computer</u> system for tracing program flow within an application<u>, the computer system comprising a memory for storing program code and a processor for processing the program code to perform the operations of</u> ~~comprising~~:
~~a user-configurable plugin module to provide options for modifying bytecode of the application at a plurality of levels of precision, each of the levels of precision specifying a particular set of methods of the application to be traced;~~

a bytecode modification module to responsively modify the bytecode at a requested level of precision responsive to a user request to modify the bytecode at the requested level of precision;

a dispatch unit to register method invocations associated with the particular set of methods specified by the level of prevision during execution of the application

providing options for modifying bytecode of the application at a plurality of levels of precision, each of the levels of precision specifying a particular set of methods of the application to be traced;

receiving a request to modify the bytecode at one of the of levels of precision;

responsively modifying the bytecode at the requested level of precision;

executing the application; and

registering method invocations associated with the particular set of methods specified by the level of prevision.

15. (Original) The system as in claim 14 wherein one of the levels of precision comprises all methods within the application.

16. (Original) The system as in claim 14 wherein one of the levels of precision comprises all methods within a package of the application.

17. (Currently amended) The system as in claim 16 wherein the application is a Java an object-oriented application and the package is a Java an object-oriented package.

18. (Original) The system as in claim 14 wherein one of the levels of

prevision comprises all methods within a particular class file of the application.

19. (Original) The system as in claim 14 wherein one of the levels of prevision comprises individually identified methods of the application.

20. (Original) The system as in claim 14 wherein, to modify the bytecode, the bytecode modification module inserts a start method invocation prior to each method of the set of methods and inserts an end method invocation following each method of the set of methods.

21. (Original) The system as in claim 20 wherein the dispatch unit stores method-related information associated with the method invocations of each of the particular set of methods.

22. (Original) The system as in claim 21 wherein the method-related information comprises an amount of time it takes for each method within the set of methods to complete.

23. (Original) The system as in claim 21 wherein the method-related information comprises a number times that each method of the set of methods is executed.

24. (Original) The system as in claim 21 wherein the method-related information comprises input and/or output parameters associated with each

method of the set of methods.

25. (Original) The system as in claim 22 further comprising tree generation logic to construct a hierarchical representation of the particular set of methods, the hierarchical representation including an indication of an amount of time it takes for each of the particular set of methods to complete.

26. (Original) The system as in claim 25 wherein the tree generation logic additionally generates a graphical tree reflecting the hierarchical representation within a graphical user interface, wherein the graphical tree includes a separate entry for each method within the set of methods.

27. (Original) An article of manufacture including program code which, when executed by a machine, causes the machine to perform the operations of:

providing options for modifying bytecode of the application at a plurality of levels of precision, each of the levels of precision specifying a particular set of methods of the application to be traced;

receiving a request to modify the bytecode at one of the of levels of precision;
responsively modifying the bytecode at the requested level of precision;
executing the application; and

registering method invocations associated with the particular set of methods specified by the level of prevision.

28. (Original) The article of manufacture as in claim 27 wherein one of the levels of precision comprises all methods within the application.

29. (Original) The article of manufacture as in claim 27 wherein one of the levels of precision comprises all methods within a package of the application.

30. (Currently amended) The article of manufacture as in claim 29 wherein the application is a Java an object-oriented application and the package is a Java an object-oriented package.

31. (Original) The article of manufacture as in claim 27 wherein one of the levels of prevision comprises all methods within a particular class file of the application.

32. (Original) The article of manufacture as in claim 27 wherein one of the levels of prevision comprises individually identified methods of the application.

33. (Original) The article of manufacture as in claim 27 wherein modifying the bytecode comprises:

inserting a start method invocation prior to each method of the set of methods and inserting an end method invocation following each method of the set of methods.

34. (Original) The article of manufacture as in claim 33 comprising additional program code which, when executed by the machine, cause the machine to perform the operations of:

storing method-related information associated with the method invocations of each of the particular set of methods.

35. (Original) The article of manufacture as in claim 33 wherein the method-related information comprises an amount of time it takes for each method within the set of methods to complete.

36. (Original) The article of manufacture as in claim 33 wherein the method-related information comprises a number times that each method of the set of methods is executed.

37. (Original) The article of manufacture as in claim 33 wherein the method-related information comprises input and/or output parameters associated with each method of the set of methods.

38. (Original) The article of manufacture as in claim 37 comprising additional program code which, when executed by the machine, cause the machine to perform the operations of:
constructing a hierarchical representation of the particular set of methods, the hierarchical representation including an indication of an amount of time it takes for each of the particular set of methods to complete.

39. (Original) The article of manufacture as in claim 38 comprising additional program code which, when executed by the machine, cause the machine to perform the operations of:
generating a graphical tree reflecting the hierarchical representation within a graphical user interface, wherein the graphical tree includes a separate entry for each method within the set of methods.